Computing Sparse Representations in O(N log N) Time

Computing Sparse Representations in $O(N \log N)$ Time

Tsung-Han Lin and H. T. Kung
School of Engineering and Applied Sciences, Harvard University
{thlin, htk}@eecs.harvard.edu

I. Introduction

Machine learning concerns forming representations of input observations to facilitate tasks such as classification. A recent insight in *deep learning* [1] is to use a deep architecture that stacks multiple levels of nonlinear operations in an inference hierarchy to extract different layers of abstractions. Deep learning is a promising direction and has attained state-of-the-art performance in some application areas such as computer vision and speech recognition. In this work, we aim to improve the computational efficiency of deep learning.

A key step in deep learning is to represent input signals as layers of sparse representations. In image processing, this means to progressively describe objects using features of larger spatial scales. For example, in the bottom layer objects can be represented by edges of different widths, lengths and orientations over small regions. In higher layers objects may be represented by shapes such as squares, triangles and so on over large regions. The representations are computed based on some designed or learned dictionary. At each layer an input signal is represented using just a few dictionary atoms.

We consider the orthogonal matching pursuit (OMP) algorithm [5], which forms sparse representations by greedily selecting representing dictionary atoms. The computation cost of OMP is proportional to the dimensions of the dictionary. Suppose that the input signal is an $M \times 1$ vector $\mathbf{x}$, and the dictionary has N $M \times 1$ atoms $\mathbf{d}_i$, $i = 1, ..., N$. The bulk of the sparse representation computation amounts to computing N correlations between $\mathbf{x}$ and $\mathbf{d}_i$ for all i. Thus the total computation cost is $O(MN)$.

Note that N is usually governed by the characteristics of a given machine learning task. For example, if the task is to classify objects with a large number of categories, N tends to be large for an increased chance of representing x with just a few dictionary atoms. On the other hand, with an easier task such as differentiating between only a few different looking objects, a relatively small N may be sufficient to derive discernible representations. M, however, is driven by the input signal size and the size of intermediate sparse representations computed in the hierarchy. This means M can be large especially in higher layers of the learning framework. For example, in image processing, sparse representations for small local regions of an image are formed in the bottom layer. The representations are then aggregated and vectorized over a larger neighborhood as the input signal for the next layer, which can easily be very long.

We show that this $O(MN)$ cost can be reduced to $O(N \log N)$, a complexity independent of the signal or representation size M. This means that the computation cost is only dictated by the desired classification resolution.

II. Performing OMP in the compressed domain

We propose to move the computations of OMP to a lower dimensional subspace, i.e., a compressed domain, thereby significantly reducing the data size and computation cost. We first state our results for a signal $\mathbf{x}$ that can be exactly represented using K atoms from a normalized $M \times N$ dictionary $\mathbf{D}$, meaning that $\mathbf{x} = \mathbf{Df}$, where $\mathbf{f}$ is a sparse vector. We define a linear mapping $\Phi : \mathbb{R}^M \rightarrow \mathbb{R}^{\widehat{M}}$ with $\widehat{M} < M$ and call the reduced-dimension problem of recovering $\mathbf{f}$ from $\Phi\mathbf{x} = \Phi\mathbf{Df}$ the *compressed domain recovery problem*.

Theorem 1. *For any given $\delta \in (0, 1)$, OMP solves the compressed domain recovery problem, where $\widehat{M}$ can be as small as $O(\log N/\delta^2)$, if the dictionary $\mathbf{D}$ is sufficiently incoherent, i.e., $\mu < \frac{1}{2K-1} - \delta$ where μ is mutual coherence of $\mathbf{D}$ defined as the maximum absolute inner product between any two atoms.*

The proof of Theorem 1 makes use of Johnson-Lindenstrauss lemma [4] that there exists a Φ which embeds the N dictionary atoms in M-dimensional space to an $O(\log N/\delta^2)$-dimensional subspace in a way that distances between atoms are nearly preserved subject to a small distortion factor δ. Consequently, the inner product between any two atoms is also preserved up to δ in the subspace [3]. This means that the coherence of the compressed dictionary $\Phi\mathbf{D}$ will not be changed by more than δ from that of $\mathbf{D}$. Theorem 1 then follows from a sufficient condition for the working of OMP that mutual coherence for the dictionary needs to be sufficiently low [5].

For constant δ and K, by Theorem 1, $\widehat{M}$ can be $O(\log N)$. Thus the computation cost of OMP is now $O(N \log N)$.

III. Compressed Domain OMP in Image Classification

We have tested compressed domain OMP for image classification. Our implementation is similar to that in [2], and has two layers. In the first layer, the sparse representations for local 6x6 patches are computed. The representations computed in layer 1 are aggregated in layer 2 over 16x16 patches and re-encoded to form higher-level sparse representations. These representations are then aggregated over the whole image to form an image-level representation. The dictionaries at layer 1 and 2 are set to have dimensions 36×108 and 2268×1000, and the sparsity K is set to 5 and 10, respectively.

We use 20 object categories from the Caltech-101 data set. After obtaining the image representations, a linear SVM is trained with 30 images for each category, and we evaluate the classification performance over 655 images. We compress the dictionary at layer 2 using random Bernoulli matrices with 2x and 10x compression ratios, giving dictionaries of size 1134×1000 and 226×1000. Without any compression, the classification accuracy is 78.9%. With 2x and 10x compression, the classification accuracy is 78% and 75%, respectively. This suggests that the computation cost of OMP can be reduced by 10x with less than 4% decrease in classification accuracy.

References

[1] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.
[2] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *NIPS*, 2011.
[3] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk. Signal processing with compressive measurements. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):445–460, 2010.
[4] W. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
[5] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, 2004.